Complexity, Systems, and Software

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Sarah A. Sheard, Ph.D.
2014

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
29 OCT 2014	**N/A**	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Complexity, Systems, and Software	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Sheard /Sarah	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**SAR**	**26**	

This material is based upon work funded and supported by FAA is sponsoring the discussion period, but the material in the presentation was created prior to my arrival at SEI (dissertation work) under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center sponsored by the United States Department of Defense.

DM-0001816

Agenda

What is complexity?

Complexity and project outcomes

Complexity of systems and software

Changing nature of systems and software

What Is Complexity?

(1) Objective–Subjective

(2) Definitions

(3) Entities

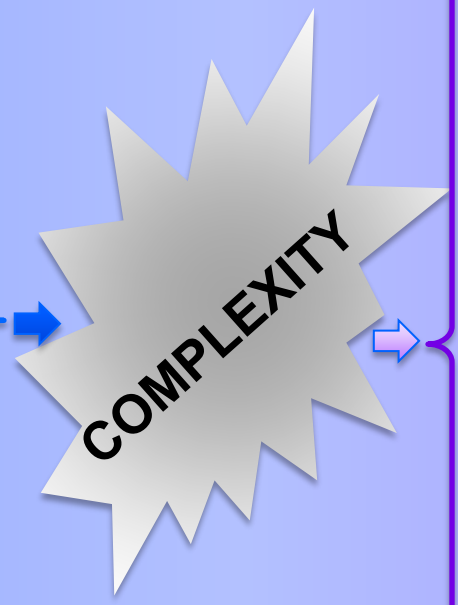
(4) Types

What Is Complexity? (1) Objective–Subjective

Software Engineering Institute

System characteristics
Technical characteristics
Objective complexity

- Many pieces
- Adaptive
- Emergent
- Nonlinear behavior
- Tightly coupled
- Self-organizing
- Decentralized
- Non-mechanical
- Chaotic behavior
- Multi-scale



Cognitive characteristics
Subjective complexity
"Perceptive" complexity

- Uncertain
- Risky
- Difficult to understand
- Difficult to predict
- Frustrating
- Uncontrollable
- Costly
- Obsolete when built
- Unclear cause/effect

What Is Complexity? (2): Definitions

Proposition: However you define complexity, your definition is incomplete

Don't call anything "complexity"

At least call it "X" complexity

Proposition: Engineering seeks complexity management; complexity reduction is one way of doing that

SysE for complexity reduction is not new

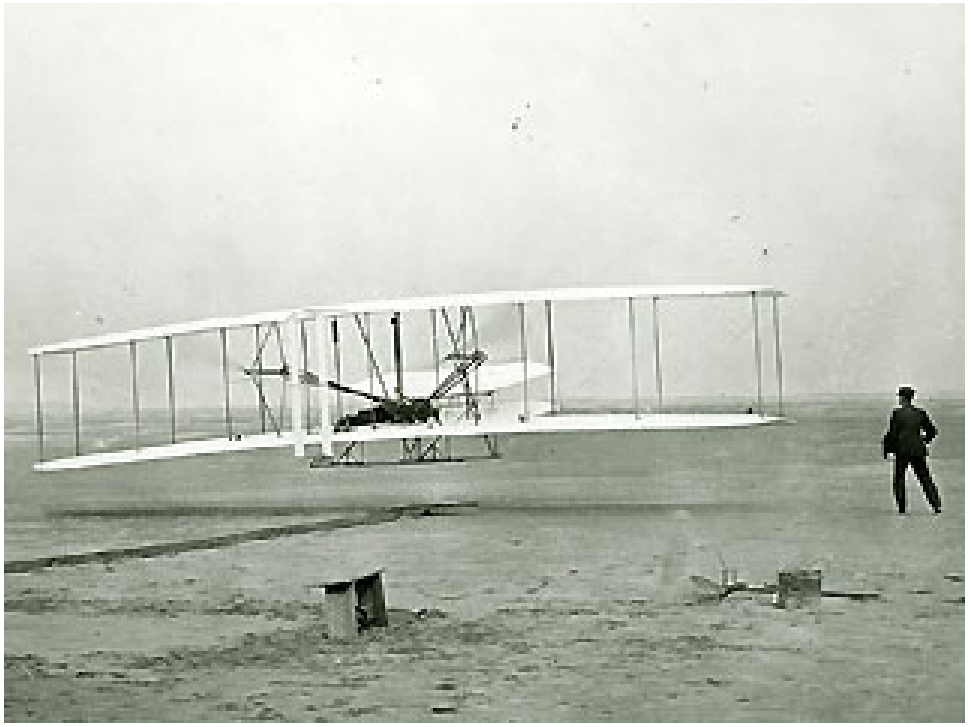
- Hall (1962): purpose of SysE is to manage complexity
- Techniques mostly not new: Complex adaptive systems, systems of systems

What Is Complexity? (2): Definitions

Complexity, defined *subjectively*, relentlessly decreases

Complexity, however defined *objectively*, relentlessly increases

Yet we manage it



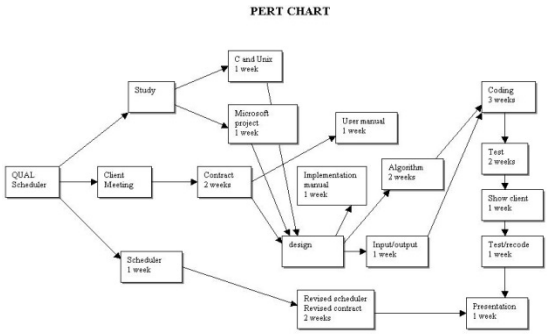
Proposition: Complexity is not a thing

… it is a characteristic of things

What Is Complexity? (3): Entities

The **system** being built

The **project** building it



The **environment** it will affect

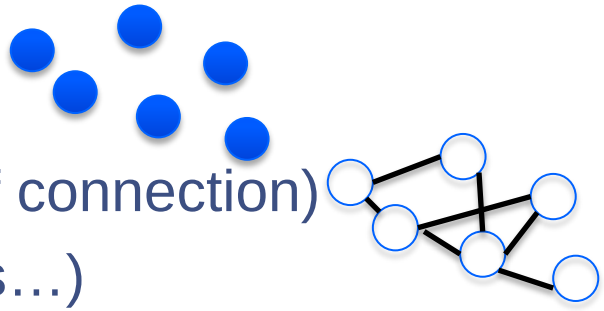
- Technical
- Socio-political

Cognitive aspects (confusion, frustration, difficulty)

What Is Complexity? (4): Types*

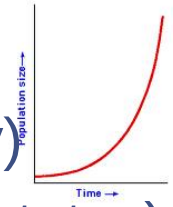
Structural

- Size (# parts, stakeholders, elements, LOC)
- Connectivity (# or density, types, strength of connection)
- Inhomogeneity (diversity, architecture, loops…)



Dynamic

- Short-term (e.g., behavioral nonlinearity)
- Long-term (evolution, transition to new states)



Socio-political

- Organizational maturity, stakeholder conflict, global context…

*(Sheard 2012)

Strike a Balance

Proposition: The point of engineering is control

Proposition: Complexity has no good side

- Study it to recognize it, to manage it, to reduce it

But: being overly simple is also wrong

Ashby's Law of Requisite Variety: A control system must have at least as many degrees of freedom as the disturbances it needs to counteract

- Technical system shouldn't be too simple (Allocating all complexity to operator)
- Technical system shouldn't be too complex (Hidden issues; dumbs down operator)

39 Complexity Questions (Sheard 2012)*

Software Engineering Institute

Subsystems
Easy, nominal, difficult requirements
Technology maturity
Architecture precedence
Schedule margin
Staff skills
Sponsors
Stakeholder conflict
Stakeholder relationships
Cognitive fog

Other questions

- Project outcomes (cost, schedule, performance, subjective assessment of outcome, produce a product)
- Project start/end dates
- Project size (cost)
- Management methods (plan, risk, agile, lean, set-based)
- Respondent role and confidence

75 programs: Did complexity correlate to cost, schedule, or performance problems?

***Sheard, Sarah A. *Assessing the impact of complexity attributes on system development project outcomes*. Dissertation, Stevens Institute of Technology, School of Systems and Enterprises, May 2012.**

Results: Top 3 Correlating Questions

Software Engineering Institute

Complexity Variable	Outcome Variable		
	Cost Overrun	Schedule Delay	Performance Shortfall
Q16d—Requirements Difficult			
Low (Under 100) group mean	3.37	3.30	2.26
High (Over 100) group mean	5.00	4.64	3.60
p-value	0.00027	0.00165	0.00163
Significance	**p<0.001**	p<0.05	p<0.05
Q32—Cognitive Fog			
Low (D-SD) group mean	3.03	2.97	2.00
High (A-SA) group mean	3.89	4.11	3.53
p-value	0.0395	0.0120	0.00074
Significance	p<0.05	p<0.05	**p<0.001**
Q38f—Stakeholder Relationships			
Low (Stable) group mean	3.30	3.11	2.15
High (Resistance) group mean	4.50	4.19	3.27
p-value	0.0209	0.0243	0.0245
Significance	p<0.05	p<0.05	p<0.05

Complexity of Systems and Software

Software: McCabe (cyclomatic) complexity: decisions in a code function

- Paths ~ edges and nodes
- Used to estimate defects & reliability

Systems: No complexity metric available

*Proposition: Measurement is inherently simplification. Measurement of complexity is like describing **Red** by means of **Green** variables*

Use knowledge of complexity:

- Identify relative complexity and relative risk
- Identify specific risks
- Identify kinds of complexity and address as risks
- Probably tie to currently collected metrics, e.g., requirements volatility

Dealing with Complexity

Determine what kind

Apply systems engineering principles and practices

Identify any special complexity as a *risk*

Study how to other fields manage that risk

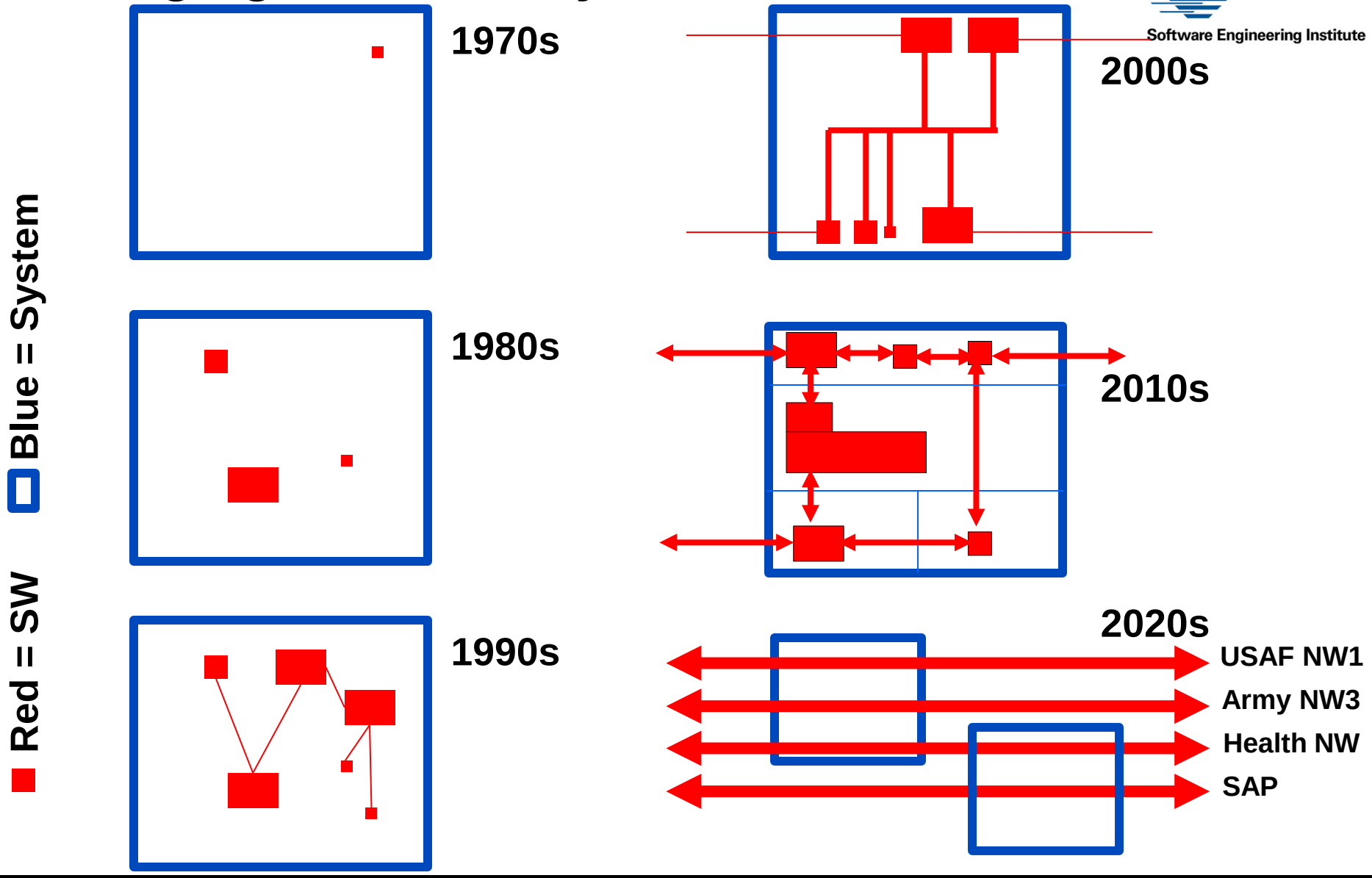
- Bring in experts

Today's "New" complexity:

Emphasis shift from "whole system" to software

- What is it?
- How should systems and software engineers manage it?

Changing Nature of Systems and Software



Conclusion

Complexity means many different things

- Countable, technical complexity vs. difficulty

Systems and software are getting ever more complex

- Complexity measures are inadequate
- Systems engineering has always been about managing complexity
- Some program characteristics predict cost & schedule problems; are they true “complexity”?

Tom Lehrer’s First Law of Thermodynamics applies

- “You can’t win, the best you can do is break even”

Contact Information

Sarah Sheard, Ph.D.
Senior Engineer
Software Solutions Division
Telephone: +1 412-268-7612
Email: sheard@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Web
www.sei.cmu.edu
www.sei.cmu.edu/contact.cfm

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257

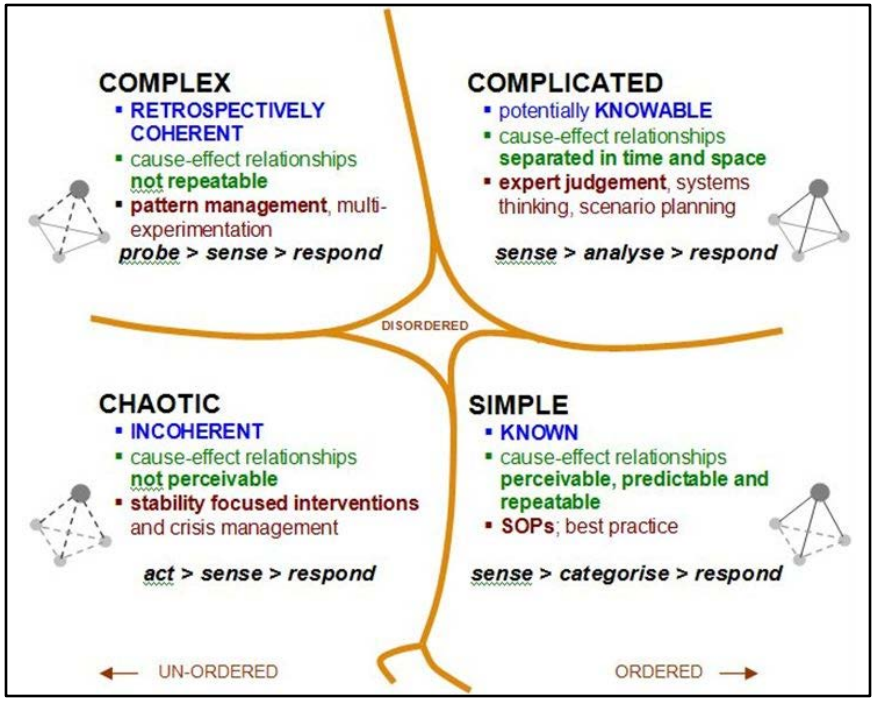
Backup Slides

Why I'm Not Talking Complex vs. Complicated

Software Engineering Institute

"Complicated" means many things

- "Can use same practices, only more of them" = MITRE (Stevens)
- Realm of systems analysis (Cynefin framework, by Kurtz and Snowden)
- Overloaded and sometimes reversed:
  - "Complexity is intrinsic, complicated is because of external influences"
  - "Complexity does not evoke difficulty; complicated refers to a high level of difficulty"



- Definitions change with time: Yesterday's complex is today's complicated, and maybe neither in the future
- Seems to be too much shorthand. "Complicated" means "what I'm not talking about" and "Complex" means "what I am talking about."

I consider "Complex" to be a spectrum

Changing Nature of Systems and Software: Needed Skills

Software Engineering Institute

T-Shaped Systems Engineer

Shallow in everything

e.g., Telemetry & Command list

Deep in something,

e.g., communications subsystem

T-Shaped Software Engineer

Very shallow in computer hardware

Moderate in all SW

Deep in own SW area

Programming, Coding, Implementation

Effectively 0 in other hardware (lubricants, mechanisms, valves)

Proposition: Software engineering = systems engineering of software system plus implementation

Complexity Questions

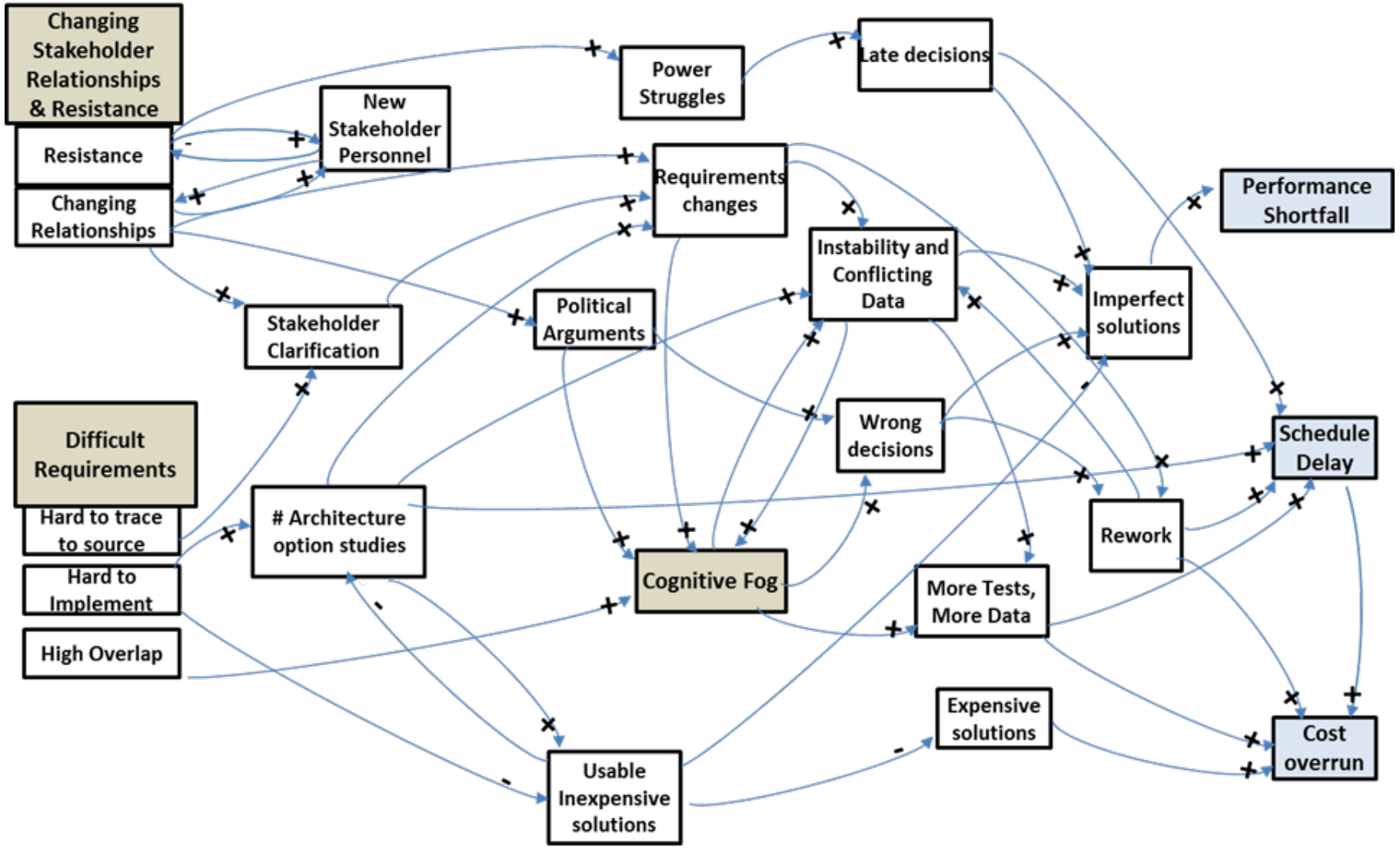
Software Engineering Institute

		Answer Choices					
#	Variable Name, Question	1	2	3	4	5	6
Complexity Variables							
16d	Requirements, Difficult Approximately how many system-level requirements did the project have initially? Difficult requirements are considered difficult to implement or engineer, are hard to trace to source, and have a high degree of overlap with other requirements. How many system requirements were there that were Difficult?	1-10	10-100	100-1000	1000-10,000	Over 10,000	
32	Cognitive Fog 'The project frequently found itself in a fog of conflicting data and cognitive overload.' Do you agree with this statement?	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	
38f	Stakeholder Relationships "Where did your project fit, on a scale of Traditional, Transitional, or Messy Frontier, in the following eight attributes?" 38f. "Stakeholder relationships: 1: Relationships stable; 2: New relationships; 3: Resistance to changing relationships.	Relation-ships stable	New Rela-tion-ships	Resist-ance to Chang-ing Rela-tion-ships			

Outcome Questions

Outcome Variables							
9	**Cost Overrun** At the point of finishing, how much did the project cost, compared to the initially predicted cost for delivery?	Under cost	At cost, +/- 5%	5-20% over plan	20-50% over	50-100% over	More than 100% over plan
10	**Schedule Delay** At the point of finishing, how long had the project taken, compared to the initially scheduled development time?	Ahead of schedule	On time within 5%	5-20% late	20-50% late	50-100% late	More than 100% late
11	**Performance Shortfall** At the point of finishing, how was the project performance, compared to the initially specified performance? (Please consider the average performance of *mission critical* features, and add any qualifiers in Notes.)	Higher than specified	Same as specified, within 5%	Low by 5-20% (fewer features or waived require-ments)	Low by 20-50%	Low by more than 50%, or project was cancelled	

One Plausible Causal Chain



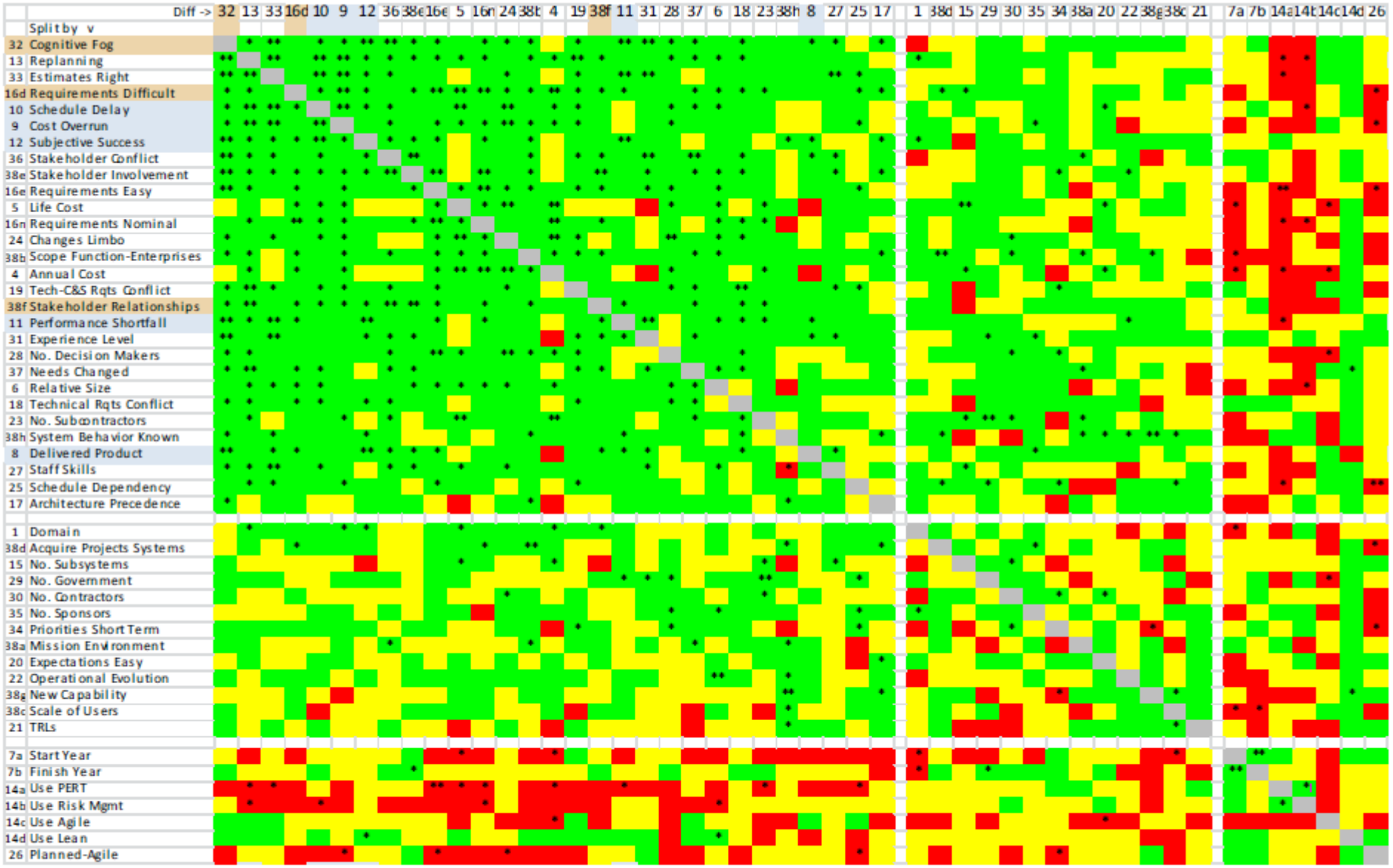


Figure 29. Congruence

*Significant, p<0.05; **Significant, p<0.001. Green: variable complexity rises together; Red: opposite; Yellow: neither. Blue=outcome variables; Beige=hypothesis variables.

What Is Complexity?

Webster: “the quality or state of being complex”

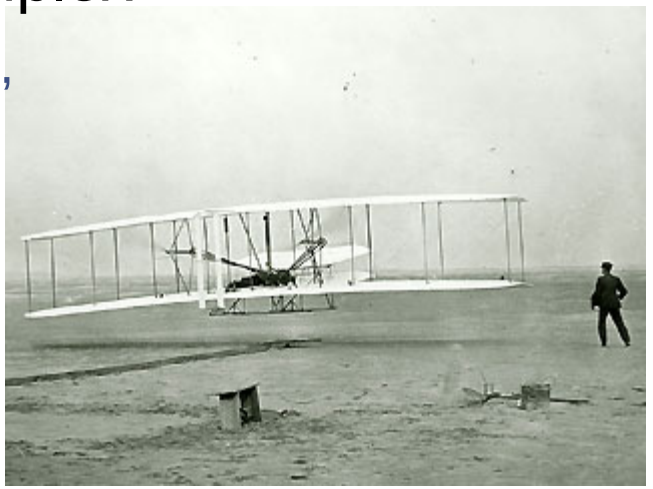
- Complex: Composite; hard to separate, analyze, or solve; concerning complex numbers

DARPA: Parts count + SLOC

Algorithmic information content

Uncertainty

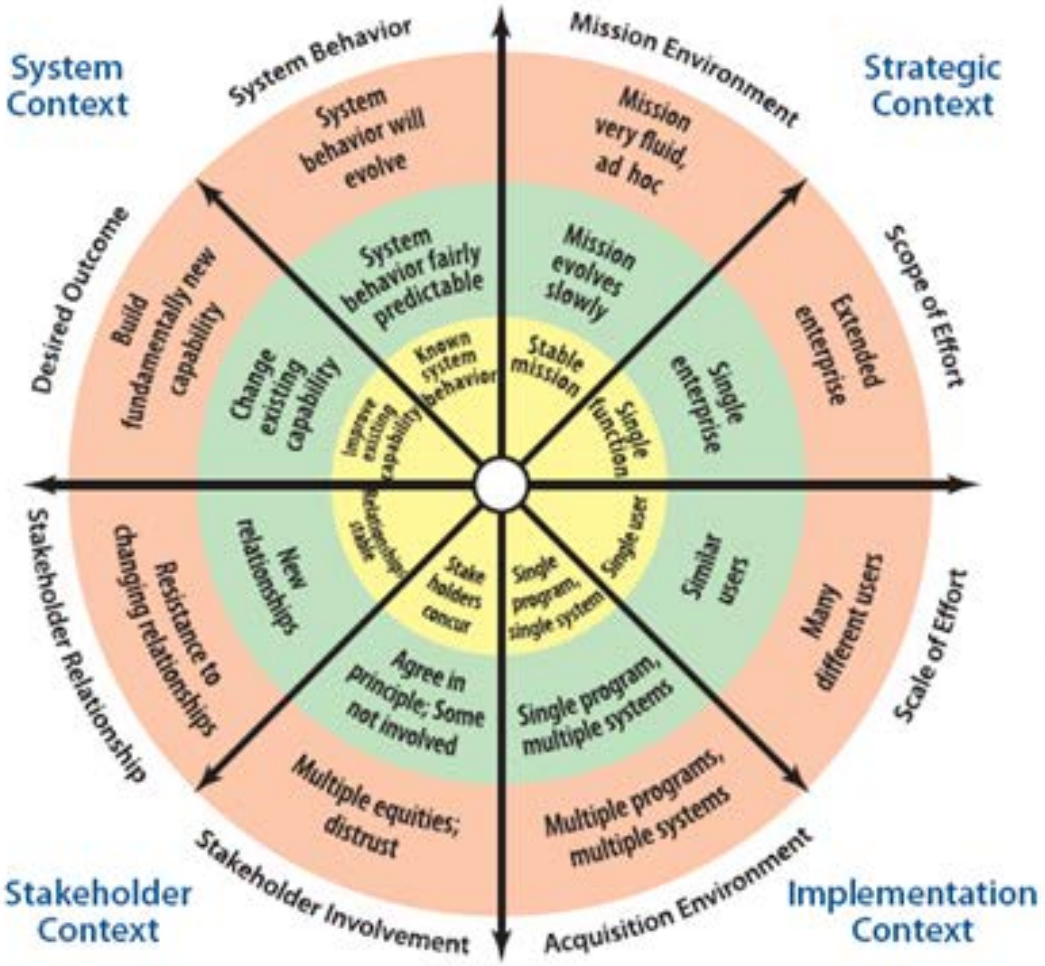
Structural, behavioral, evaluative, nested



Automated conflict avoidance for aircraft traversing airspace boundaries at different and changing altitudes and speeds, avoiding weather, considering all stakeholders have varying financial interests…

Little guidance for systems engineering

Addressing Complexity in SoSs



Traditional program domain
- *Well-bounded problem*
- *Predictable behavior*
- *Stable environment*

Transitional domain
- *Systems engineering across boundaries*
- *Influence vs. authority*

Messy frontier
- *Political engineering (power, control …)*
- *High risk, potentially high reward*
- *Foster cooperative behavior*

Source: SEBOK Wiki